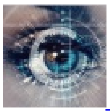
2021DASCTF八月挑战赛Writeup

原创

塞纳河畔的春水 于 2021-08-31 10:47:41 发布 2436 收藏 7
分类专栏： CTF_Writeup 文章标签： 网络安全 python pycharm github 数据仓库


CTF_Writeup 专栏收录该内容
8 篇文章 2 订阅
订阅专栏

文章目录

看大佬们都不想写这个wp那我写一个吧Orz

MISC

签到

看看公告

> flag{welcome_to_dasctf_aug}

寒王'sblog

好家伙，不知道gitee是什么根本找不到。直接在url后面加上/flag.jpg访问仓库：

```
https://hanwang2333.gitee.io/2020/03/12/outguess/flag.jpg
```

拿到flag.jpg，然后根据寒王博客里的outguess解密拿到flag

stealer

打开流量包，过滤DNS，发现有很多重复的数据，过滤ip

```
dns and ip.src==172.27.221.13
```

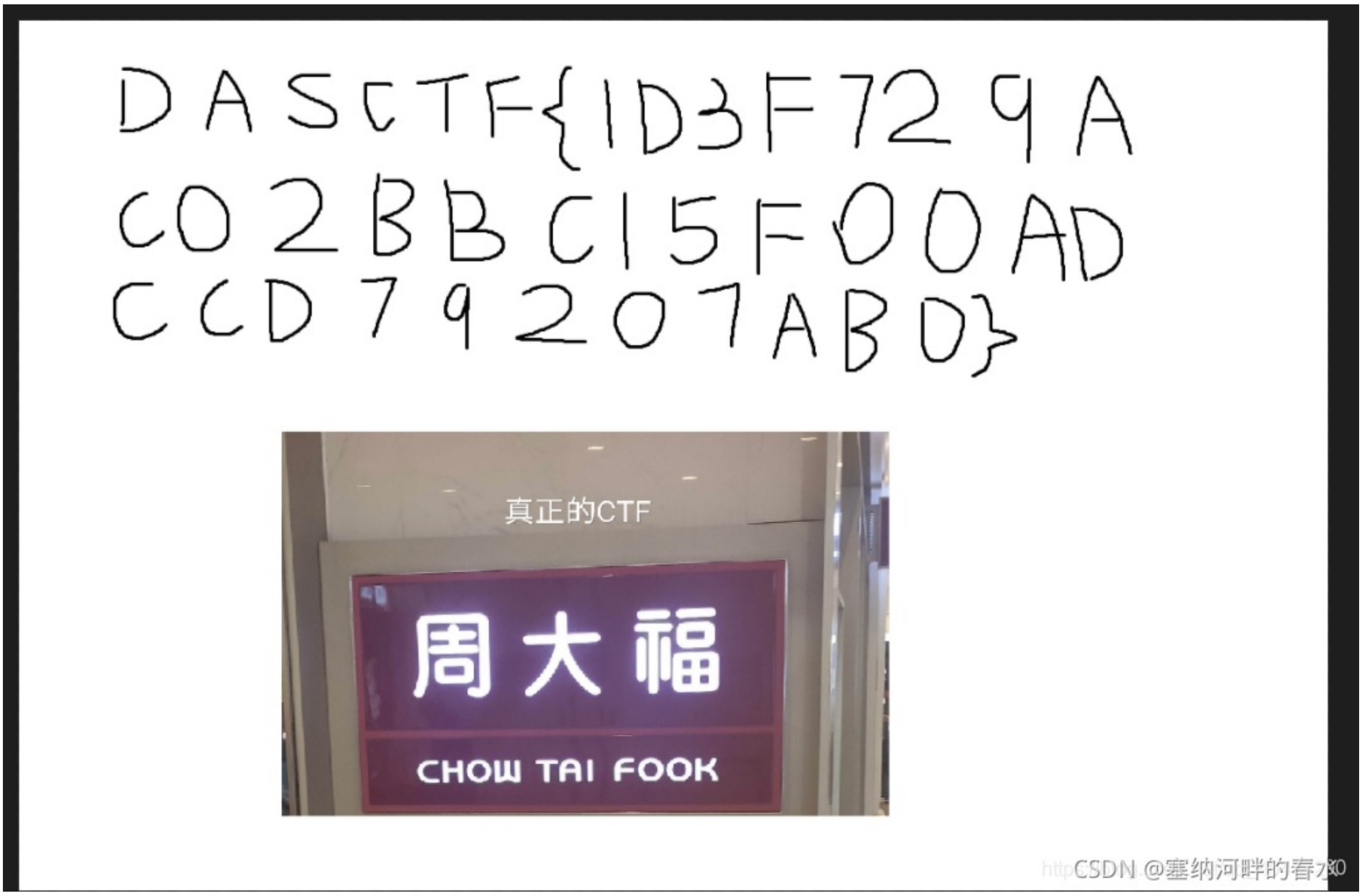
将info取出，观察发现是图片的base64编码，将字符串进行编辑方便转码。

```
字符串的变化如下：
原字符串：
Standard query 0x6a7a A iVBORw0KGgoAAAANSUhEUgAABMoAAAMxCAIAAACVY8g6AAAAAXNSR0IAr-.s4c6QAAAARnQU1BAACxjwv8Y

操作：
1、去除多余字符串”Standard query 0x6a7a A”、”ctf.com.cn OPT”、”-.”
2、将“*”替换为“+”

转化后字符串：iVBORw0KGgoAAAANSUhEUgAABMoAAAMxCAIAAACVY8g6AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwA
```

拿到图片



> 1d3f729ac02bbc15f00adccd79207ab0

CRYPTO

easymath

题目：

```
assert(len(open('flag.txt', 'rb').read()) < 50)
assert(str(int.from_bytes(open('flag.txt', 'rb').read(), byteorder='big') << 10000).endswith(
    '1862790884563160582365888530869690397667546628710795031544304378154769559410473276482265448754388655981
```

好家伙，这不TSGCTF原题，放弃思考，上大佬详细WP。

let's play with rsa~

题目：

```
from sympy import isprime,nextprime
from Crypto.Util.number import getPrime as getprime ,long_to_bytes,bytes_to_long,inverse
flag='flag{**************}'

def play():
    p=getprime(1024)
    q=getprime(1024)

    n=p*q
    e=65537

    print "Hello,let's play rsa~\n"
    print 'Now,I make some numbers,wait a second\n'
    n1=getprime(200)
    n2=getprime(200)
    number=n1*n2
    print "Ok,i will send two numbers to you,one of them was encoded.\n"
    print "Encode n1:%d,\n"%(pow(n1,e,n))
    print "And n2:%d.\n"%n2

    print "Information that can now be made public:the public key (n,e):(%d,%d)\n"%(n,e)
    while True:
        try:
            c=int(raw_input("ok,now,tell me the value of the number (encode it for safe):"))
        except:
            print "Sorry,the input is illeagal, and the integer is accept~"
        else:
            break
    d=inverse(e,(p-1)*(q-1))
    m=pow(c,d,n)
    if m==number:
        print "It's easy and interesting,didn't it?\n"
        print "This is the gift for you :"+flag
    else:
        print "Emmmmm,there is something wrong, bye~\n"

if __name__ == '__main__':
    play()
```

思路

> 题目给出n、e、pow(n1,e,n)、n2，求c
>
> $c = number^e\%n = (n1*n2)^e\%n = ((n1^e\%n)*(n2^e\%n))\%n$

```
#n =
#n2 =
#e = 65537
a = pow(n1,e,n) #题目给出
c = (a * pow(n2,e,n)) % n
print(c)
#提交c及返回flag
```

ezRSA

题目：

```
from secret import flag
from Crypto.Util.number import *
from random import getrandbits
from hashlib import sha256


class EzRsa:
    def __init__(self):
        self.E = 0x10001
        self.P = getPrime(1024)
        self.Q = getPrime(1024)
        while GCD((self.P-1)*(self.Q-1), self.E) != 1:
            self.Q = getPrime(1024)
        self.N = self.P*self.Q

    def encrypt(self):
        f = getrandbits(32)
        c = pow(f, self.E, self.N)
        return (f, c)

    def encrypt_flag(self, flag):
        f = bytes_to_long(flag)
        c = pow(f, self.E, self.N)
        return c


def proof():
    seed = getrandbits(32)
    print(seed)
    sha = sha256(str(seed).encode()).hexdigest()
    print(f"sha256({seed>>18}...).hexdigest() = {sha}")
    sha_i = input("plz enter seed: ")
    if sha256(sha_i.encode()).hexdigest() != sha:
        exit(0)


if __name__ == "__main__":
    proof()
    print("welcome to EzRsa")
    print("""
    1. Get flag
    2. Encrypt
    3. Insert
    4. Exit
    """)
    A = EzRsa()
    coin = 5
    while coin > 0:
        choose = input("> ")
        if choose == "1":
            print(
                f"pow(flag,e,n) = {A.encrypt_flag(flag)}\ne = 0x10001")
            exit(0)
        elif choose == "2":
            f, c = A.encrypt()
            print(f"plain = {f}\ncipher = {c}")
            coin -= 1
        elif choose == "3":
            q = getrandbits(1024)
            n = A.P*q
            f = getrandbits(32)
            c = pow(f, 0x10001, n)
            print(f"plain = {f}\ncipher = {c}")
            coin -= 1
        elif choose == "4":
            print("bye~")
        else:
            print("wrong input")
    print("Now you get the flag right?")
```

思路：给你5个coin，相当于四次选择信息2、3的机会，当然是平均分配啦。得到四组f、c，两组同q解n，两组不同q解p。

> 计算n：
>
> $$\begin{cases} f_1^e \ \% \ (p*q) = c_1 \\ f_2^e \ \% \ (p*q) = c_2 \end{cases}$$
> $$\Rightarrow \begin{cases} f_1^e - c_1 = k_1 * p * q \\ f_2^e - c_2 = k_2 * p * q \end{cases}$$
> $$\Rightarrow gcd(f_1^e - c_1, f_2^e - c_2) = p * q = n$$
>
> 计算q:
>
> $$\begin{cases} f_3^e \ \% \ (p*q_3) = c_3 \\ f_4^e \ \% \ (p*q_4) = c_4 \end{cases}$$
> $$\Rightarrow \begin{cases} f_3^e - c_3 = k_3 * p * q_3 \\ f_4^e - c_4 = k_4 * p * q_4 \end{cases}$$
> $$\Rightarrow gcd(f_3^e - c_3, f_4^e - c_4) = p$$

p、q、n、c都知道了，常规RSA解密。

REVERSE

py

题目简单粗暴，直接给py.exe，ida打开shift+F12搜索查看到pyinstaller字样（也可通过图标判断）认定为pyinstaller打包，直接exe转pyc。

```
python pyinstxtractor.py pay.exe
```

拿到py.pyc，上uncompyle6反编译。

```
uncompyle6 py.pyc > py.py
```

拿到py代码，简单的异或操作，写脚本进行逆运算输出flag。

```
def encode(s):
    str = ''
    for i in range(len(s)):
        res = ord[s[i]] ^ 32
        res += 31
        str += chr(res)
    return str

def decode(s):
    str = ''
    for i in range(len(s)):
        res = ord(s[i])-31
        res ^= 32
        str += chr(res)
    return str


m = 'ek`fz13b3c5e047b`bd`0/c268e600e7c5d1`|'

#strings = ''
#strings = input(‘Input:’)

print(decode(m))

#if encode(strings) == m:
#    print('Correct!')
#else:
#    print('Try again!')

#flag{24c4d6f158cacea10d379f711f8d6e2a}
```